Decentralised Data Fusion with Parzen Density Estimates

Matthew Ridley, Ben Upcroft, Lee Ling (Sharon) Ong, Suresh Kumar, Salah Sukkarieh

ARC Centre of Excellence in Autonomous Systems (CAS)
The University of Sydney
NSW Australia
$\{$*m.ridley,b.upcroft,s.ong,s.kumar,s.sukkarieh*$\}$*@cas.edu.au*

Abstract

Decentralised sensor networks typically consist of multiple processing nodes supporting one or more sensors. These nodes are interconnected via wireless communication.

Practical applications of Decentralised Data Fusion have generally been restricted to using Gaussian based approaches such as the Kalman or Information Filter. This paper proposes the use of Parzen window estimates as an alternate representation to perform Decentralised Data Fusion. It is required that the common information between two nodes be removed from any received estimates before local data fusion may occur. Otherwise, estimates may become overconfident due to data incest. A closed form approximation to the division of two estimates is described to enable conservative assimilation of incoming information to a node in a decentralised data fusion network.

A simple example of tracking a moving particle with Parzen density estimates is shown to demonstrate how this algorithm allows conservative assimilation of network information.

1. Introduction

There are a variety of techniques for alternate representions of probability distributions to improve the performance of tracking algorithms for non-linear and/or non-Gaussian sensors and or systems. Such approaches include the Particle Filter[4], Gaussian Mixtures[1], [5], [10], Support Vector Machines[2] or even discretised distributions[14]

Decentralised Data Fusion (DDF) systems such as those developed by Nettleton[8] have remained exclusively tied to the Gaussian distributions of the Information Filter. A method to decentralise a Particle Filter is described by Rosencrantz[11].

The Multivariate Gaussian distribution of the state $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{\Sigma}$ as used in the majority of tracking (both centralised and decentralised) applications is defined for reference:

$$\mathbf{P}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}]^T \mathbf{\Sigma}^{-1}[\mathbf{x}-\bar{\mathbf{x}}]} \tag{1}$$

2. Generalised Decentralised Data Fusion

Figure 1 depicts the major components and interactions of a decentralised sensing node. Each sensor node processes raw sensor data to generate a likelihood. This is then fused with a

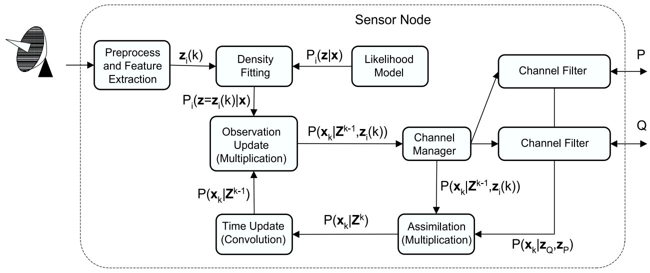

Fig. 1: General Decentralised Node

local estimate. This local estimate is then transmitted to other nodes to be fused similarly. The channel filter is responsible for maintaining an estimate of the information a connected node has in common with itself. The assimilation process requires the removal of common information and fusion of new information. Communicating information increments, while possible, is not robust to communication failure.

A. The Channel Filter

This is the area of primary concern as removal of common information from communicated distributions other than a single Gaussian is non-trivial. Use of the Covariance Intersect algorithm (for Gaussian distributions) as described in [6] provides a means to generate conservative updates between multiple decentralised nodes. This process eliminates the need for a channel filter but is typically more conservative than would be desired.

Given Bayes Rule:

$$\mathbf{P}(\mathbf{x}|\mathbf{Z}^k) = \frac{\mathbf{P}(\mathbf{x}|\mathbf{Z}^{k-1})\mathbf{P}(\mathbf{z}(k)|\mathbf{x})}{\mathbf{P}(\mathbf{z}(k)|\mathbf{Z}^{k-1})} \tag{2}$$

It can shown that if a channel filter maintains the common information between two nodes a and b $\mathbf{P}(\mathbf{x}|\mathbf{Z_a} \cap \mathbf{Z_b})$ and node b then transmits its new state to node a then the Bayesian channel update at node a now becomes:

$$\mathbf{P}(\mathbf{x}|\mathbf{Z}) \propto \frac{\mathbf{P}(\mathbf{x}|\mathbf{Z_a})\mathbf{P}(\mathbf{x}|\mathbf{Z_b})}{\mathbf{P}(\mathbf{x}|\mathbf{Z_a} \cap \mathbf{Z_b})} \tag{3}$$

As mentioned previously, a division is required to remove the common information held between two nodes.

3. GAUSSIAN PARZEN WINDOW ESTIMATES

General kernel based methods of representing probability distributions are described by Parzen[9]. While in general any kernel may be used, the Gaussian kernel remains the most useful. This is due to the fact that most operations are closed in form, and are therefore more efficient.

A parzen density estimate (with a gaussian kernel) is defined as:

$$\mathbf{P}(\mathbf{x}) = \sum_{i=1}^{n} \gamma_i G_i(\mathbf{x}) \tag{4}$$

where $G_i(\mathbf{x})$ is a Gaussian distribution on $\mathbf{x}$ and γ_i is a weight where $\sum_{i=1}^{n} \gamma_i = 1$

Parzen density estimates with Gaussian kernels can be included in the family of Gaussian sum representations. Their use in tracking applications appears in work by Alspach and Sorenson [1], [13], where there is suggestion that using a single Gaussian kernel with multiple means and weights significantly simplifies the problem.

Gaussian mixtures typically use multiple different kernels. However, storage of multiple kernels consumes substantially more space than that of additional sample means alone. When only a single kernel is used, the storage and operations on multiple kernels may be eliminated at the expense of requiring more kernels to adequately represent the distribution. Overall this may significantly reduce the required communication bandwidth within a sensor network.

A. Product of two Gaussian Parzen density estimates

If we have two parzen density estimates i,j with n_i, n_j Gaussian samples respectively, their product will be:

$$\begin{aligned}
\mathbf{P}_{ij}(\mathbf{x}) &= \mathbf{P}_i(\mathbf{x})\mathbf{P}_j(\mathbf{x}) \\
&= \sum_{i=1}^{n_i} \gamma_i G_i(\mathbf{x}) \sum_{j=1}^{n_j} \gamma_j G_j(\mathbf{x}) \\
&= \sum_{i=1}^{n_{ij}} \gamma_{ij} G_{ij}(\mathbf{x})
\end{aligned} \tag{5}$$

where $n_{ij} = n_i * n_j$ and $G_{ij}(\mathbf{x})$ is the product of Gaussian $G_i(\mathbf{x})$ with $G_j(\mathbf{x})$. The resulting increase in sample size can be a cause of concern if not carefully dealt with.

B. Parzen Estimate Sample Reduction

After an update operation resulting in a large increase of sample, a practical method of reducing this sample size is required. Approaches by Girolami[3] and Robert[10](using EM) to reduce or condense the number of samples have been proposed, but performance is not suitable for real-time tracking applications. Suitable methods need to be efficient but also effective enough so as not to excessively distort the original distribution.

C. Information Measures for Parzen Estimates

A measure of information content of a distribution is important for both communication and sensor management. Entropy, considered as the only reasonable measure of informativeness[12] is difficult to compute for Parzen density estimates. However the Quadratic Renyi Entropy measure provides an analytical solution for the family of sum of Gaussian distributions. Torkkola [15] suggests such approaches, consequently it is the measure used for evaluation of results in Section 6-D. Renyi Entropy does not provide a direct measure of information, but does preserve ordering of like distributions. For a parzen density estimate the Renyi Entropy is simply the log of the sum of the resultant weights from the square of the original estimate $\sum_{i=1}^{n} \gamma_{ii} G_{ii}$

$$\begin{aligned}
H_{R(2)}(\mathbf{P}(\mathbf{x})) &= -log \int \mathbf{P}(\mathbf{x})^2 d\mathbf{x} \\
&= -log \int \left(\sum_{i=1}^{n} \gamma_i G_i(\mathbf{x}) \right)^2 d\mathbf{x} \\
&= -log \int \sum_{i=1}^{n} \gamma_{ii} G_{ii}(\mathbf{x}) d\mathbf{x} \\
&= -log \left(\sum_{i=1}^{n} \gamma_{ii} \right)
\end{aligned} \tag{6}$$

4. BASIC GAUSSIAN OPERATIONS

Some basic Gaussian operations will be defined for the purpose of understanding the division operation.

A. Product of two n dimensional Gaussians

The product of two distributions is important for performing Bayesian updates on distributions.

The product of two multivariate gaussians in n dimensions can be shown to be:

$$\begin{aligned}
\mathbf{P}_i(\mathbf{x})\mathbf{P}_j(\mathbf{x}) &= \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_i]^T \mathbf{\Sigma}_i{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_i]} \\
&\quad \times \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_j|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_j]^T \mathbf{\Sigma}_j{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_j]} \\
\mathbf{P}_{ij}(\mathbf{x}) &= k_{ij} \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_{ij}|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_{ij}]^T \mathbf{\Sigma}_{ij}{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_{ij}]}
\end{aligned} \tag{7}$$

where

$$k_{ij} = \frac{|\mathbf{\Sigma}_{ij}|^{1/2}}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}|\mathbf{\Sigma}_j|^{1/2}} e^{-\frac{1}{2}\left[\bar{\mathbf{x}}_i^T \mathbf{\Sigma}_i{}^{-1}\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_j^T \mathbf{\Sigma}_j{}^{-1}\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_{ij}^T \mathbf{\Sigma}_{ij}{}^{-1}\bar{\mathbf{x}}_{ij}\right]} \tag{8}$$

and

$$\mathbf{\Sigma}_{ij}{}^{-1} = \mathbf{\Sigma}_i{}^{-1} + \mathbf{\Sigma}_j{}^{-1} \tag{9}$$

and

$$\bar{\mathbf{x}}_{ij} = \mathbf{\Sigma}_{ij}\left(\mathbf{\Sigma}_i{}^{-1}\bar{\mathbf{x}}_i + \mathbf{\Sigma}_j{}^{-1}\bar{\mathbf{x}}_j\right) \tag{10}$$

B. Division of two Gaussians in n dimensions

Just as Multiplication provides new information to a state estimate, division removes information. In the decentralised context, this is regularly occurring within the channel filter.

$$\frac{\mathbf{P}_i(\mathbf{x})}{\mathbf{P}_j(\mathbf{x})} = \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_i]^T \mathbf{\Sigma}_i{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_i]} \times \frac{(2\pi)^{n/2}|\mathbf{\Sigma}_j|^{1/2}}{1} e^{\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_j]^T \mathbf{\Sigma}_j{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_j]} \quad (11)$$

$$\mathbf{P}_{ij}(\mathbf{x}) = k_{ij} \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_{ij}|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_{ij}]^T \mathbf{\Sigma}_{ij}{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_{ij}]} \quad (12)$$

where

$$k_{ij} = \frac{|\mathbf{\Sigma}_{ij}|^{1/2}|\mathbf{\Sigma}_j|^{1/2}}{(2\pi)^{n/2}|\mathbf{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}\left[\bar{\mathbf{x}}_i^T \mathbf{\Sigma}_i{}^{-1}\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j^T \mathbf{\Sigma}_j{}^{-1}\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_{ij}^T \mathbf{\Sigma}_{ij}{}^{-1}\bar{\mathbf{x}}_{ij}\right]} \quad (13)$$

and

$$\mathbf{\Sigma}_{ij}{}^{-1} = \mathbf{\Sigma}_i{}^{-1} - \mathbf{\Sigma}_j{}^{-1} \quad (14)$$

and

$$\bar{\mathbf{x}}_{ij} = \mathbf{\Sigma}_{ij}\left(\mathbf{\Sigma}_i{}^{-1}\bar{\mathbf{x}}_i - \mathbf{\Sigma}_j{}^{-1}\bar{\mathbf{x}}_j\right) \quad (15)$$

with the condition that $\mathbf{\Sigma}_{ij}{}^{-1}$ is Symmetric Positive Definite. ie. a valid covariance matrix.

C. Derivative of an n dimension Gaussian

For the division approximation we need the gradient of the surface of the original distribution. Maybeck[7] shows how such derivatives are performed.

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}}\mathbf{P}(\mathbf{x}) &= \frac{\partial}{\partial \mathbf{x}} \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}]^T \mathbf{\Sigma}^{-1}[\mathbf{x}-\bar{\mathbf{x}}]} \\ &= -\frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}]^T \mathbf{\Sigma}^{-1}[\mathbf{x}-\bar{\mathbf{x}}]} \\ &\quad \times [\mathbf{x}-\bar{\mathbf{x}}]^T \mathbf{\Sigma}^{-1} \\ &= -\mathbf{P}(\mathbf{x})[\mathbf{x}-\bar{\mathbf{x}}]^T \mathbf{\Sigma}^{-1} \end{aligned} \quad (16)$$

D. Gradient of a Gaussian Parzen density estimate

Consequently we can see that if a Parzen estimate is a sum of Gaussians, then the gradient of a Parzen estimate is the sum of derivatives of each of these Gaussians.

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{P}(\mathbf{x}) = -\sum_{i=1}^{n} \gamma_i G_i(\mathbf{x})[\mathbf{x}-\bar{\mathbf{x}}_i]^T \mathbf{\Sigma}^{-1} \quad (17)$$

5. Closed Form Approximation to Parzen Density Estimate Division

This section will detail an approach to provide a conservative approximation to the division of two Parzen density estimates.

A. Justification

The division of a sum of gaussians by another does not result in a closed form solution. An approximation is therefore the only possible solution. This resulting approximation of new information would also have to be conservative, to avoid any chance of data incest occurring.

B. Proposed Method

The division may be decomposed into a sum of division operations comprised of each weighted gaussian of the original divedend, divided by the original divisor. The divisor is then locally approximated (by a single weighted gaussian) at the mean of the current weighted gaussian forming the divedend. A division of a Gaussian by a Gaussian is possible (with some restrictions). For the Parzen density estimator case, the kernel chosen for the approximation is the kernel of the original divisor. This results in all quotients then having the same kernel, thereby maintaining form.

In order to determine the mean and weight of the divisor approximator, both the divisor and it's gradient must be evaluated at the point of interest (the mean of the current divedend). The kernel of the approximator may be directly obtained, and is the same for all divisions.

C. Derivation

The decomposition into a sum of division operations may be seen below.

$$\begin{aligned}\frac{\mathbf{P}_a(\mathbf{x})}{\mathbf{P}_b(\mathbf{x})} &= \frac{\sum_{i=1}^{N_a} \gamma_i \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_a|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_i]^T \mathbf{\Sigma}_a{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_i]}}{\sum_{j=1}^{N_b} \gamma_j \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_b|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_j]^T \mathbf{\Sigma}_b{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_j]}} \\ &= \frac{\gamma_1 \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_a|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_1]^T \mathbf{\Sigma}_a{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_1]}}{\sum_{j=1}^{N_b} \gamma_j \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_b|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}-\bar{\mathbf{x}}_j]^T \mathbf{\Sigma}_b{}^{-1}[\mathbf{x}-\bar{\mathbf{x}}_j]}} \\ &\quad + \ldots \end{aligned} \quad (18)$$

The divisor approximator for each i with mean $\tilde{\mathbf{x}}_i$ and covariance Σ_b is developed accordingly.

$$\tilde{\mathbf{P}}_i(\mathbf{x}) = \tilde{\gamma}_i \tilde{G}_i(\mathbf{x}) \quad (19)$$

To determine the mean and weight of the divisor approximator $\tilde{\mathbf{P}}_i(\mathbf{x})$ the divisor and it's gradient is evaluated for each weighted Gaussian $\gamma_i G_i(\mathbf{x})$ at $\bar{\mathbf{x}}_i$. This will require order N_b operations for each i. From Equation 16 we may solve for the mean of the approximating gaussian. Since the value of the divisor and approximator are equal, and their respective gradients are also equal at the point of interest $\bar{\mathbf{x}}_i$ it follows that:

$$\mathbf{P}_b(\bar{\mathbf{x}}_i) = \tilde{\mathbf{P}}_i(\bar{\mathbf{x}}_i) \quad (20)$$

and

$$\frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i) = \frac{\partial}{\partial \mathbf{x}}\tilde{\mathbf{P}}_i(\bar{\mathbf{x}}_i) \quad (21)$$

Solving for the mean of the approximator:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i) &= -\tilde{\mathbf{P}}_i(\bar{\mathbf{x}}_i)[\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i]^T \mathbf{\Sigma}_b{}^{-1} \\ \frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i) &= -\mathbf{P}_b(\bar{\mathbf{x}}_i)[\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i]^T \mathbf{\Sigma}_b{}^{-1} \\ \frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i)\mathbf{\Sigma}_b &= -\mathbf{P}_b(\bar{\mathbf{x}}_i)[\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i]^T \\ \frac{\frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i)\mathbf{\Sigma}_b}{-\mathbf{P}_b(\bar{\mathbf{x}}_i)} &= [\bar{\mathbf{x}}_i - \tilde{\mathbf{x}}_i]^T \\ \tilde{\mathbf{x}}_i &= \bar{\mathbf{x}}_i + \left[\frac{\frac{\partial}{\partial \mathbf{x}}\mathbf{P}_b(\bar{\mathbf{x}}_i)\mathbf{\Sigma}_b}{\mathbf{P}_b(\bar{\mathbf{x}}_i)}\right]^T \end{aligned} \quad (22)$$

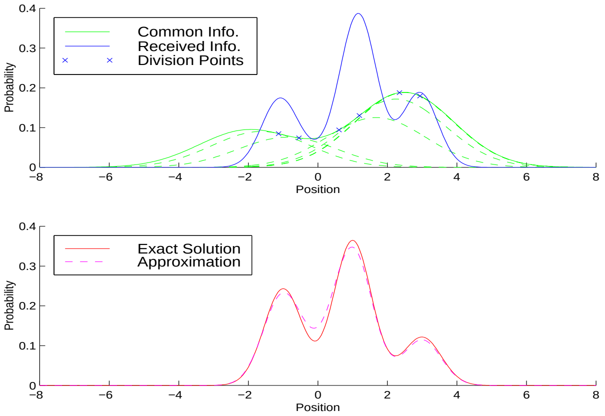

Fig. 2: Single dimension example of division. Crosses represent the points of division. Dashed lines show the approximators used.

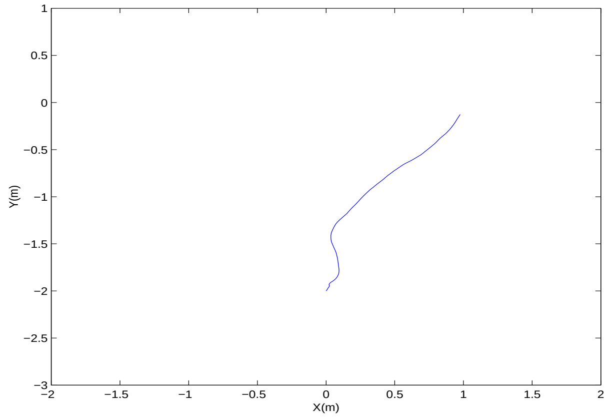

Fig. 3: Path of Particle

Evaluating this new Gaussian at $\bar{\mathbf{x}}_i$ will allow us to determine the required weight.

$$\tilde{\gamma}_i = \frac{\mathbf{P}_b(\bar{\mathbf{x}}_i)}{\tilde{G}(\bar{\mathbf{x}}_i)} \tag{23}$$

From this point forward, the division operation as defined in Section 4-B, proceeds for each $G_i(\mathbf{x})$. The resultant weights from the divisions must then be normalised to provide a total sum of 1. The overall operation is a $N_a \times N_b$ operation, with an output of N_a samples. This is more convenient than the multiplication, as no resampling operation is required.

D. A Single Dimension Example

Figure 2 shows an example division of two distributions representing the situation where common information is removed from an incoming distribution. The "New Information" may then be fused with the local estimate and re-communicated.

In this example the entropy of the exact solution is 1.7065, whilst the approximation has an entropy of 1.7137. This indicates that the approximation of new information may be safely used without fear of any data incest occurring. The approximation is visibly "flatter" but still preserves the important features of the original observation.

6. A Simulation in 4 Dimensions

For an example we will track a single particle exhibiting random walk behaviour. The particle path is shown in Figure 3 Two bearing-range sensors of differing characteristics will be employed. Sensor 1 has good bearing accuracy, while sensor 2 has superior range accuracy. These two sensors will form a 2 node DDF network.

A. Likelihood Fitting

The sensor observations and their respective likelihood functions are approximated by a Parzen density estimate. If the likelihood function is also a valid probability density function, then this may be sampled from and reduced using methods similar to that described in Section 3-B. A more economical approach is to eqi-space kernels strategically located on and around the modes of the likelihood, with the weight of each kernel being awarded the value of the likelihood function at that point. The shape of the kernel is chosen to reflect the general shape of the underlying distribution. An example of this process is shown in Figure 5 having been applied to the likelihood function in Figure 4. This reflects the likelihood of sensor 1.

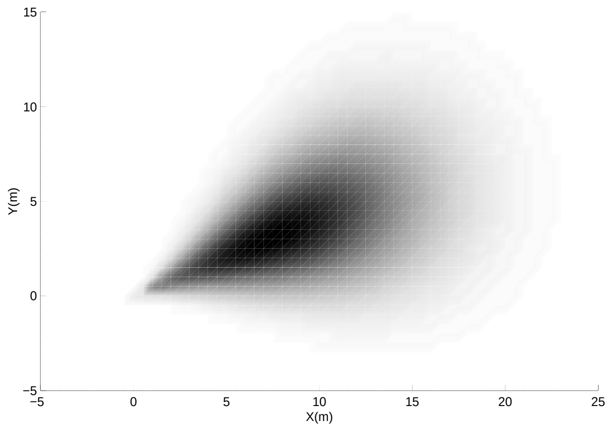

Fig. 4: Original likelihood

B. Motion Model

The Integrated Ornstein-Uhlenbeck process as defined by Stone[14] was used to model the particle as it allows for bounding the brownian velocity over time. This prevents wayward measurements inducing an excessively large velocity that persists whilst the target can no longer be observed.

The velocity is bounded by appropriate choice of the model parameter γ. This can be expressed as:

$$\mathbf{x}(k) = \mathbf{F}_{(k,k-1)}\hat{\mathbf{x}}(k-1 \mid k-1) + \mathbf{w}_{(k,k-1)} \tag{24}$$

where the state vector is

$$\mathbf{x}(k) = [x(k), \dot{x}(k), y(k), \dot{y}(k)]^T \tag{25}$$

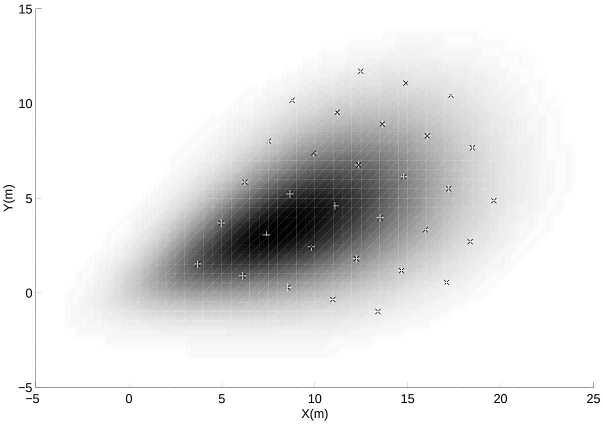

Fig. 5: Likelihood approximation. Crosses mark kernel centres.

The state transition matrix for this system is given by

$$\mathbf{F}_{(k,k-1)} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & F_v & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & F_v \end{bmatrix} \tag{26}$$

where

$$F_v = e^{-\Delta T \gamma} \tag{27}$$

The process is $\mathbf{G}_k \mathbf{Q}_{(k,k-1)} \mathbf{G}_k^T$ where

$$\mathbf{Q}_{(k,k-1)} = \begin{bmatrix} q_x & 0 \\ 0 & q_y \end{bmatrix} \tag{28}$$

and

$$\mathbf{G}_k = \begin{bmatrix} 0 & 0 \\ \sqrt{\Delta T}(1 - F_v) & 0 \\ 0 & 0 \\ 0 & \sqrt{\Delta T}(1 - F_v) \end{bmatrix} \tag{29}$$

Time evolution of a Parzen density estimate, with a simple linear Gaussian model involves considering each sample as in individual Kalman Filter. More complicated models using Parzen density estimates are possible, but will result in multiple kernels requiring additional sample reduction or re-estimation of the distribution.

C. Sample Reduction

The sample reduction (or condensation) method employed in this example simply maintains the N highest weighted samples after an update. This behaves like a Multi-Hypothesis Filter. This approach however tends to preserve peaks at the expense of any tails that may exist in the distribution. The covariance of the kernel is adjusted to preserve the covariance of the original distribution to reduce some of the effect of the distortion.

D. Results

Figures 6 through 9 provide a snapshot of the distributions at the end of the simulation. They provide visual confirmation of correct DDF operation. The DDF Solution of Node 2 was omitted as it was similar to the results of Node 1, which depicts a final distribution similar to, but slightly less compact than that of the centralised solution.

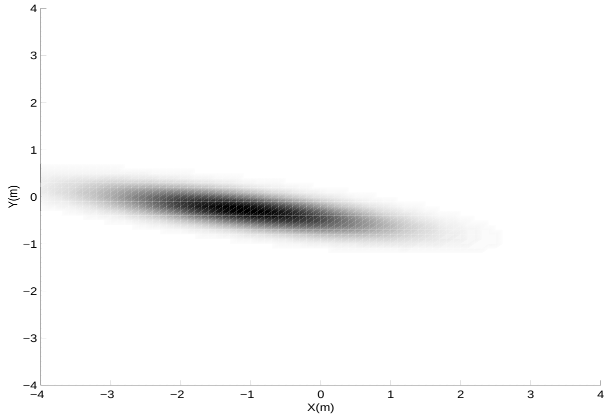

Fig. 6: Node 1 operating independently

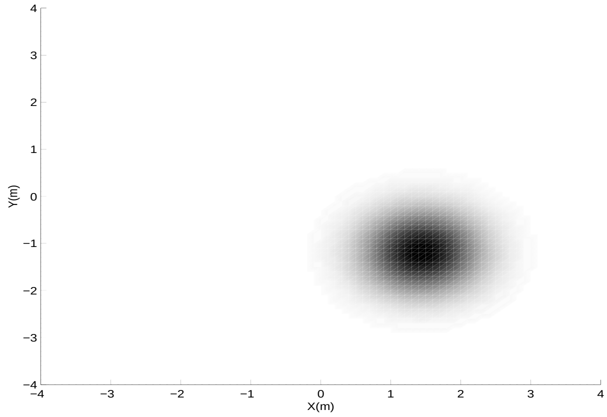

Fig. 7: Node 2 operating independently

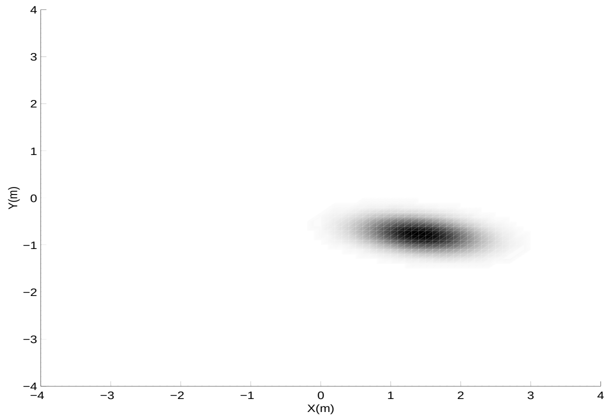

Fig. 8: Node 1 with DDF

As can be seen from both Figure 10, (the spatial distance between the mean of the estimate and the actual position) and Figure 11, the decentralised nodes exhibit performance close to that of the centralised case, and also perform better that the sensor operating alone.

In order to evaluate whether the DDF updates are conservative and not causing data incest, one would expect the

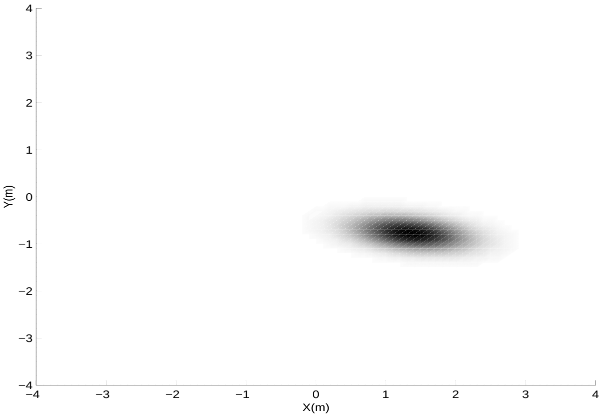

Fig. 9: Centralised solution

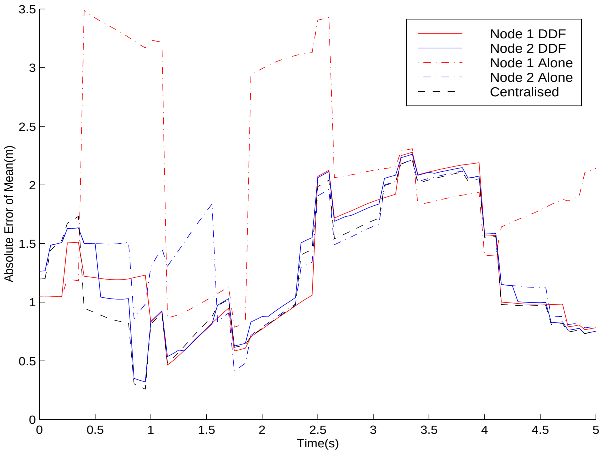

Fig. 10: Absolute error of mean of solutions

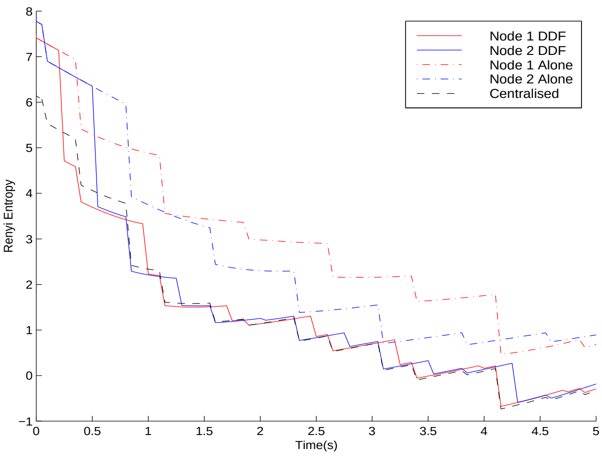

Fig. 11: Renyi entropy of solutions

DDF nodes to have solutions with a larger entropy. Figure 11 shows the result. The transitions in entropy are caused by local observations and channel updates. Whilst the steady state solution for the centralised solution has the least Renyi Entropy, the decentralised nodes indicate periods of non-conservative behaviour. However, the crude resampling scheme employed to reduce the number of kernels after each update has some significant disadvantages which would be familiar to those using particle filters (such as sampling impoverishment). These distorting effects would account for this behaviour as the DDF node would cull different kernels to the centralised solution.

7. Conclusion

Using the conservative division approximation algorithm described in this paper, we now have an effective means to perform decentralised Bayesian estimation with more elaborate probability distribution representations. Given the results there exists a need for an efficient but more effective condensation (refitting) algorithm. Further expansion of the algorithm to include more general Gaussian mixture distributions would also be advantageous.

Acknowledgements

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[1] Daniel L. Alspach and Harold W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, August 1972.

[2] S. Challa, M. Palaniswami, and A. Shilton. Distributed data fusion using support vector machines. In *Proceedings of The Fifth International Conference on Information Fusion*, pages 881–885, July 2002.

[3] Mark Girolami and Chao He. Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, 25(10):1253–1264, October 2003.

[4] N. J. Gordon, D. J. Salmon, and A. F. M. Smith. A novel approach to nonlinear/nongaussian bayesian state estimation. In *IEEE Proceedings on Radar Signal Processing.*, pages 107–113, 1993.

[5] Kazufumi Ito and Kaiqi Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, May 2000.

[6] Simon Julier and Jeffrey Uhlmann. A nondivergent estimation algorithm in the presence of unknown correlations. In *Proceedings of The American Control Conference, Vol 4*, pages 2369–2373. IEEE, July 1997.

[7] P. Maybeck. *Stochastic Models, Estimation and Control, Volume 1*. Academic Press Inc, New York, 1979.

[8] E. Nettleton and H. Durrant-Whyte. Delayed and asequent data in decentralised sensing networks. In G.T. McKee and P.S. Schenker, editors, *Sensor Fusion and Decentralised Control in Robotic Systems IV*, volume 4571, pages 1–9, 2001.

[9] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, September 1962.

[10] Christian P. Robert. Mixtures of distributions: inference and estimation. In W. Gilks, S. Richardson, and D. Spiegelhalter, editors, *Markov chain Monte Carlo in practice*, Interdiscip. Statist., chapter 24, pages 441–464. Chapman & Hall, London, 1996.

[11] M. Rosencrantz, G. Gordon, and S. Thrun. Decentralized sensor fusion with distributed particle filters. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico, 2003.

[12] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, July and October 1948.

[13] H. W. Sorenson and D. L. Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7:465–479, 1971.

[14] Lawrence D. Stone, Carl A. Barlow, and Thomas L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.

[15] Kari Torkkola. Feature extraction by non parametric mutual information maximization. *The Journal of Machine Learning Research*, 3:1415–1438, March 2003.